METHOD FOR THE EXCHANGE OF DATA BETWEEN CONTROLS OF MACHINES, PARTICULARLY ROBOTS

FIELD OF THE INVENTION

The invention relates to a method for the exchange of data between controls of machines, particularly robots.

BACKGROUND OF THE INVENTION

The transmission of data between controls of machines at present only takes place by means of direct links, either in the form of directly wired inputs and outputs or by means of data transmission field bus systems, while commands or instructions cannot be transmitted in this way. Commands must be given by means of operator elements or a mainframe computer. The corresponding data are transmitted as such directly between the two controls.

The hitherto known procedure is complicated, because physical inputs/outputs or the field bus system must be made available. Coincidence must be brought about on both controls, so that the inputs/outputs have the same significances. As the controls do not normally communicate directly via a field bus, but instead by means of an interposed plant programmable logic control (PLC), the latter must take account of a 1:1 mapping in the inputs/outputs. If communication between the controls is to be extended, this involves a change to the control programs of the input and output configuration (additional field bus words, addi-

tional lines for physical inputs/outputs, additional programming expenditure on the PLC side).

The problem of the invention is to overcome the indicated disadvantages and in particular permit a simplification and simple extendibility of communication between two or more controls, whilst largely eliminating fault-prone intermediate layers.

SUMMARY OF THE INVENTION

According to the invention the set problem is solved with a method of the aforementioned type, wherein a first control produces an instruction to be transmitted with data to be sent to a second control and with an identification representing said second control, wherein the instruction to be transmitted is provided with an identification of the first control, wherein the first control sends the instruction to be transmitted to the second control, wherein the second control evaluates the data of the instruction and wherein the second control supplies an acknowledgement to the first control.

The invention provides a single-line instruction which, apart from the actual data to be transmitted (physical states and/or control commands) comprises an identification of the receiver control. This creates simple extendibility of the communication by freely programmable commands as parameters of the control instruction (string commands), so that an increased transparency is possible in the user program by the use of freely programmable commands as parameters of the instructions and therefore the use of corresponding names for the commands instead of I/O designations. The invention makes extendibility very simple, because it is only necessary to declare a new global vari-

able, which can then be used on the receiver control, because the latter can manipulate a random global variable. Extendibility is not limited by hardware resources. Apart from data, i.e. in particular physical states, it is possible to also transmit commands in this way. As a result of the invention data and command transmission to robots can take place flexibly without hardware/software expenditure on exchanging robots and without fixing a hierarchy, which is a prerequisite for direct cooperation of balanced robots. A further advantage is that the transmission takes place synchronously, so that the transmission initiator gains a direct acknowledgement concerning the result of the action and can only continue to operate when the command is performed on the other side. According to the invention, not only simple data can be transmitted, by means of a value allocation, but in targeted manner system states can be manipulated, such as the forcing of a step operating mode or the performance of a set selection in the other program.

According to a preferred development of the invention, the command to be transmitted is formatted as a UDP message and in particular the acknowledgement of the second control is also formatted as a UDP message. According to further developments of the invention, the second control compiles the data received in an internal code and in particular, if the transmitted data contain a control command, the second control executes the same. Finally, according to the invention, the first control only transmits an instruction to be transmitted to a further control on receiving therefrom an acknowledgement to the effect that the second control is ready to perform an instruction. The instruction to be transmitted can also contain the identification of the control in question and also the data to be transmitted in the form of a constant or variable.

BRIEF DESCRIPTION OF THE DRAWINGS

Further advantages and features of the invention can be gathered from the claims and the following description of an embodiment of the invention with reference to the attached drawings, wherein show:

Fig. 1 A diagrammatic representation of several mutually cooperating controls.

Fig. 2 The fundamental sequence of the method according to the invention for exchanging data between two controls.

Fig. 3 A more precise representation of the processing of the data to be exchanged between controls in the same.

DETAILED DESCRIPTION OF THE DRAWINGS

Fig. 1 shows in exemplified manner four cooperating controls 1 to 4 with each of which is associated a clear address, IP="192.0.1.1" for control 1. Each control 1 to 4 has a computer 1.1 on which runs a control program for a machine, particularly for a robot controlled by the particular control. Each control also has an ethernet driver 1.2 to 4.2 and a network card 1.3 to 4.3 for communication with a network 5, by means of which the controls 1 to 4 are interconnected directly or by means of a switch. The memories contained in a control, such as read-only memories, volatile memories, etc., as well as peripherals, such as in particular input and output devices, are not shown.

For data exchange purposes between two controls 1 and 2, initially a control program running on the first control produces an instruction to be transmitted, which on the one hand includes the data to be transmitted, such as physical parameters of the robot associated with the first control or also instructions for modifying physical parameters in a robot associated with the second control. The instruction to be transmitted also includes the address of the control to which the data or instructions produced by the program running on the first computer are to be transmitted. In addition, said instruction is provided with the sender's own address. This message is then transmitted as a UDP message via the network to the second control (fig. 2). The latter receives the UDP message and then compiles the command to an internal control code. In the embodiment of fig. 2, the interpreter of the control software running in the control 2 executes the command transmitted to it, namely I=I+1, i.e., increase by a value of "1" the physical quantity I. Then the control 2 produces an acknowledgement information, which on the one hand contains the address of the first control and on the other a statement concerning the performance result of the received and executed instruction. This message is sent back as a UDP message from control 2 to control 1, whose interpreter waits for this command response, received by the first control and evaluated by the control program of the first control and as a result of the evaluation optionally further instructions are sent by the first control to the second control.

In this and also in the example of fig. 3 the control 1 acts as a server, whereas control 2 (or also control 3) operates as a client. Associations are not fixed and can instead be modified as a function of the tasks to be performed.

Before the control 1 sends a command to another control, control 2 and/or 3 in fig. 3, control 1 initially waits for controls 2 and 3 to perform their tasks, optionally based on a remote control command and give a corresponding acknowledgement, as explained in connection with fig. 2. Then a new command sequence is transmitted to the controls, namely to weld and to fetch a part.

Further remote control commands can follow. The control then again waits until the controls 2 and 3 (clients 1 and 2) have given an acknowledgement about the performance of the transmitted responses.

Controls 2 and 3 (clients 1 and 2) wait for a task, execute the commands received (weld, fetch) and then return to the idle/waiting state.

The address of the particular control can be used as a constant (e.g. RemoteCmd ("192.0.1.3", ...)) or as a variable (e.g. IP[]="192.0.1.2" RemoteCmd(IP[], ...)). The same applies regarding the data to be transmitted, which can also be fully used as a constant (RemoteCmd(..., "MyVar=44")) or a variable (CmdString[]="MyVar=055", RemoteCmd(..., CmdString[])).

Standard commands are the selection of a program on the control in question (RUNProgName()), the resetting of a program (RESET), the cancellation of a program (CANCEL) and the allocation of a value to a global variable (value allocation) as simple values, strings or Boolean values or also in the form of complex allocations, such as sums, smaller/larger estimates and complex functions. Finally, a further possible command involves the waiting for a specific system state for the controlled control, such as e.g.

the value of an input 1 becoming equal to that of an input 2: RemoteCmd(IP[], "Wait for $IN[1]==$IN[2]").

LIST OF REFERENCE NUMERALS

1-4	Controls
1.1	Computer
1.2-4.2	Ethernet drivers
1.3-4.3	Network cards
5	Network